Communication-Efficient Decentralized Learning

Yuejie Chi

Carnegie Mellon University

EdgeComm Workshop, 2020
Acknowledgements

Boyue Li
CMU

Shicong Cen
CMU

Yuxin Chen
Princeton

Distributed/Federated learning: due to privacy and scalability, data are distributed at multiple locations / workers / agents.

Let \( M = \bigcup_i M_i \) be a data partition with equal splitting:

\[
f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x), \quad \text{where} \quad f_i(x) := \frac{1}{(N/n)} \sum_{z \in M_i} \ell(x; z).
\]

- \( N = \) number of total samples
- \( n = \) number of agents
- \( N/n = \) number of local samples \( m \)
Decentralized ERM - algorithmic framework

\[ \text{minimize}_{x} \quad f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \]

\[ \Downarrow \]

\[ \text{minimize}_{x_i} \quad \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) \quad \text{subject to} \quad x_i = x_j \]
Decentralized ERM - algorithmic framework

\[
\minimize_{\bm{x}} \quad f(\bm{x}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\bm{x})
\]

\[
\downarrow
\]

\[
\minimize_{\bm{x}_i} \quad \frac{1}{n} \sum_{i=1}^{n} f_i(\bm{x}_i) \quad \text{subject to} \quad \bm{x}_i = \bm{x}_j
\]

- **Local computation**: agents update local estimate;

  \[
  \Rightarrow \text{need to be scalable!}
  \]
Decentralized ERM - algorithmic framework

\[
\begin{align*}
\text{minimize}_x & \quad f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \\
\Downarrow & \\
\text{minimize}_{x_i} & \quad \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) \quad \text{subject to} \quad x_i = x_j
\end{align*}
\]

- **Local computation**: agents update local estimate;
  \[\Rightarrow \text{need to be scalable!}\]

- **Global communications**: agents exchange for consensus;
  \[\Rightarrow \text{need to be communication-efficient!}\]
Decentralized ERM - algorithmic framework

\[
\begin{align*}
\text{minimize}_{x} & \quad f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \\
\downarrow \\
\text{minimize}_{x_i} & \quad \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) \quad \text{subject to} \quad x_i = x_j
\end{align*}
\]

- **Local computation**: agents update local estimate;
  \[\Rightarrow \text{need to be scalable!}\]

- **Global communications**: agents exchange for consensus;
  \[\Rightarrow \text{need to be communication-efficient!}\]

**Guiding principle**: more local computation leads to less communication.
Two distributed schemes

Master/slave model

PS coordinates global information sharing
Two distributed schemes

Master/slave model
PS coordinates global information sharing

Network model
agents share local information over a graph topology
Distributed first-order methods in the master/slave setting

\[ x_i^t \leftarrow \text{LocalUpdate}(f_i, \nabla f(x^t), x^t) \]

\[ \nabla f(x^t) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x^t) \]

\[ x^t = \frac{1}{n} \sum_{i=1}^{n} x_{i}^{t-1} \]

- Parameter consensus
- Gradient consensus
- Local data

Distributed Approximate NEwton (DANE) (Shamir et. al., 2014):

\[ x^t_i = \arg\min_{x} f_i(x) - \langle \nabla f_i(x^t_{i-1}) - \nabla f_i(x^{t-1}), x \rangle + \mu \frac{1}{2} \| x - x^{t-1} \|^2 \]

- Quasi-Newton method and less sensitive to ill-conditioning.

Distributed Stochastic Variance-Reduced Gradients (Cen et. al., 2020):

\[ x_{t,s}^i \leftarrow x_{t,s}^{i-1} - \eta v_{t,s}^{i-1} \]

- Variance-reduced stochastic gradient, \( s = 1, 2, \ldots \)
- Better local computation efficiency.
Distributed first-order methods in the master/slave setting

\[ \mathbf{x}_i^t \leftarrow \text{LocalUpdate}(f_i, \nabla f(\mathbf{x}^t), \mathbf{x}^t) \]

\[ \nabla f(\mathbf{x}^t) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x}^t) \]

\[ \mathbf{x}^t = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^{t-1} \]

local data

parameter consensus

gradient consensus

Quasi-Newton method and less sensitive to ill-conditioning.

Distributed Stochastic Variance-Reduced Gradients (Cen et al., 2020):

\[ \mathbf{x}_{t,s}^i \leftarrow \mathbf{x}_{t,s}^{i-1} - \eta \mathbf{v}_{t,s}^{i-1} \]

\[ \mathbf{v} \text{ variance-reduced stochastic gradient} \]

Better local computation efficiency.
Distributed first-order methods in the master/slave setting

\[ \mathbf{x}_i^t \leftarrow \text{LocalUpdate}(f_i, \nabla f(\mathbf{x}^t), \mathbf{x}^t) \]

\[ \nabla f(\mathbf{x}^t) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x}^t) \]

\[ \mathbf{x}^t = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^{t-1} \]

local data

gradient consensus

parameter consensus

Distributed Approximate NEwton (DANE) (Shamir et. al., 2014):

• Quasi-Newton method and less sensitive to ill-conditioning.

Distributed Stochastic Variance-Reduced Gradients (Cen et. al., 2020):

• Better local computation efficiency.
Distributed first-order methods in the master/slave setting

\[ \mathbf{x}_i^t \leftarrow \text{LocalUpdate}(f_i, \nabla f(\mathbf{x}^t), \mathbf{x}^t) \]

\[ \nabla f(\mathbf{x}^t) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(\mathbf{x}^t) \]

\[ \mathbf{x}^t = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_{i}^{t-1} \]

local data

gradient consensus

parameter consensus

Distributed Approximate NEwton (DANE) (Shamir et al., 2014):

- Quasi-Newton method and less sensitive to ill-conditioning.

Distributed Stochastic Variance-Reduced Gradients (Cen et al., 2020):

- Better local computation efficiency.
Distributed first-order methods in the master/slave setting

\[ \mathbf{x}_i^t \leftarrow \text{LocalUpdate}(f_i, \nabla f(\mathbf{x}^t), \mathbf{x}^t) \]

\[ \nabla f(\mathbf{x}^t) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x}^t) \]

\[ \mathbf{x}^t = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^{t-1} \]

**Distributed Approximate NEwton (DANE)** (Shamir et. al., 2014):

\[ \mathbf{x}_i^t = \arg\min_{\mathbf{x}} f_i(\mathbf{x}) - \langle \nabla f_i(\mathbf{x}^{t-1}) - \nabla f(\mathbf{x}^{t-1}), \mathbf{x} \rangle + \frac{\mu}{2} \| \mathbf{x} - \mathbf{x}^{t-1} \|^2 \]

- Quasi-Newton method and less sensitive to ill-conditioning.
Distributed first-order methods in the master/slave setting

\[ \mathbf{x}_i^t \leftarrow \text{LocalUpdate}(f_i, \nabla f(\mathbf{x}^t), \mathbf{x}^t) \]

\[ \nabla f(\mathbf{x}^t) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x}^t) \]

\[ \mathbf{x}^t = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}^{t-1}_i \]

Distributed Stochastic Variance-Reduced Gradients (Cen et al., 2020):

\[ \mathbf{x}^{t,s}_i \leftarrow \mathbf{x}^{t,s-1}_i - \eta \mathbf{v}^{t,s-1}_i, \quad s = 1, 2, \ldots \]

- Better local computation efficiency.

\[ \mathbf{x}^{t,s}_i \leftarrow \mathbf{x}^{t,s-1}_i - \eta \mathbf{v}^{t,s-1}_i, \quad s = 1, 2, \ldots \]

\[ \mathbf{x}^{t,s}_i \leftarrow \mathbf{x}^{t,s-1}_i - \eta \mathbf{v}^{t,s-1}_i, \quad s = 1, 2, \ldots \]
Naive extension to the network setting

\[ f_1(x) \]
\[ f_2(x) \]
\[ f_3(x) \]
\[ f_4(x) \]
\[ f_5(x) \]

\{ x^t_i, \nabla f_i(x^t_i) \}

• **Communicate:** agent transmits \{x^t_i, \nabla f_i(x^t_i)\};

• **Compute:**

\[ x^t_i \leftarrow \text{LocalUpdate}(f_i, \text{Avg}\{\nabla f_j(x^t_j)\}_{j \in N_i}, \text{Avg}\{x^t_j\}_{j \in N_i}) \]

\text{surrogate of } \nabla f(x^t) \quad \text{surrogate of } x^t
Naive extension to the network setting

- **Communicate:** agent transmits \( \{x_i^t, \nabla f_i(x_i^t)\} \);
- **Compute:**

\[
x_i^t \leftarrow \text{LocalUpdate}(f_i, \text{Avg}\{\nabla f_j(x_j^t)\}_{j \in \mathcal{N}_i}, \text{Avg}\{x_j^t\}_{j \in \mathcal{N}_i})
\]

== surrogate of \( \nabla f(x^t) \) == surrogate of \( x^t \)

Doesn't converge to global optimum!
Naive extension to the network setting

- **Communicate:** agent transmits $\{x_i^t, \nabla f_i(x_i^t)\}$;
- **Compute:**
  
  $$x_i^t \leftarrow \text{LocalUpdate}(f_i, \text{Avg}\{\nabla f_j(x_j^t)\}_{j \in N_i}, \text{Avg}\{x_j^t\}_{j \in N_i})$$

  surrogate of $\nabla f(x^t)$  
  surrogate of $x^t$

Consensus needs to be designed carefully in the network setting!
Average dynamic consensus

Assume that each agent generates some time-varying quantity $r^t_j$.

How to track its the dynamic average $rac{1}{n} \sum_{j=1}^{n} r^t_j = \frac{1}{n} \mathbf{1}_n^\top \mathbf{r}^t$ in each of the agents, where $\mathbf{r}^t = [r^t_1, \cdots, r^t_n]^\top$?
Average dynamic consensus

Assume that each agent generates some time-varying quantity $r_j^t$.

How to track its the dynamic average $\frac{1}{n} \sum_{j=1}^{n} r_j^t = \frac{1}{n} 1_n^\top r^t$ in each of the agents, where $r^t = [r_1^t, \cdots, r_n^t]^\top$?

- Dynamic average consensus (Zhu and Martinez, 2010):

$$q^t = W q^{t-1} + \underbrace{r^t - r^{t-1}}_{\text{correction}},$$

where $q^t = [q_1^t, \cdots, q_n^t]^\top$ and $W$ is the mixing matrix.
Average dynamic consensus

Assume that each agent generates some time-varying quantity \( r_j^t \).

How to track its the dynamic average \( \frac{1}{n} \sum_{j=1}^{n} r_j^t = \frac{1}{n} 1_n^\top r^t \) in each of the agents, where \( r^t = [r_1^t, \cdots, r_n^t]^\top \)?

- Dynamic average consensus (Zhu and Martinez, 2010):
  \[
  q^t = \underbrace{W q^{t-1}}_{\text{mixing}} + \underbrace{r^t - r^{t-1}}_{\text{correction}},
  \]
  where \( q^t = [q_1^t, \cdots, q_n^t]^\top \) and \( W \) is the mixing matrix.

- **Key property:** the average of \( \{q_i^t\} \) dynamically tracks the average of \( \{r_i^t\} \);
  \[
  1_n^\top q^t = 1_n^\top r^t,
  \]

---

Gradient tracking

\[ x_i^t \leftarrow \text{LocalUpdate}(f_i, \nabla f(x^t), x^t) \]

- Parameter averaging:
  \[ y_j^t = \sum_{k \in \mathcal{N}_j} w_{jk} x_{k}^{t-1}, \]

- Gradient tracking:
  \[ s_j^t = \sum_{k \in \mathcal{N}_j} w_{jk} s_{k}^{t-1} + \left( \nabla f_j(y_j^t) - \nabla f_j(y_j^{t-1}) \right). \]
Gradient tracking

\[ x_i^t \leftarrow \text{LocalUpdate}(f_i, \nabla f(x^t), x^t) \]

- Parameter averaging:
  \[ y_j^t = \sum_{k \in N_j} w_{jk} x_k^{t-1}, \]

- Gradient tracking:
  \[ s_j^t = \sum_{k \in N_j} w_{jk} s_k^{t-1} + \nabla f_j(y_j^t) - \nabla f_j(y_j^{t-1}) \]

We can now apply the same DANE and SVRG-type local updates!
Linear Regression: Well-Conditioned

\[ f_i(x) = \|y_i - A_i x\|_2^2, \quad A_i \in \mathbb{R}^{1000 \times 40} \]

**Figure:** The optimality gap w.r.t. iterations and gradients evaluation. The condition number \( \kappa = 10 \). ER graph \((p = 0.3)\), 20 agents.
Linear Regression: Ill-Conditioned

\[ f_i(x) = \| y_i - A_i x \|_2^2, \quad A_i \in \mathbb{R}^{1000 \times 40} \]

**Figure:** The optimality gap w.r.t. iterations and gradients evaluations. The condition number \( \kappa = 10^4 \). ER graph \( (p = 0.3) \), 20 agents.
The mixing rate of the graph $\alpha_0 = 0.922$. A single round of mixing within each iteration cannot ensure the convergence of Network-SVRG.
The mixing rate of the graph $\alpha_0 = 0.922$. A single round of mixing within each iteration cannot ensure the convergence of Network-SVRG.
The mixing rate of the graph $\alpha_0 = 0.922$. A single round of mixing within each iteration cannot ensure the convergence of Network-SVRG.
Final remarks

• gradient tracking provides a way to extend master/slave algorithms (DANE and SVRG) to network settings;
• probing computational-communication trade-offs by employing different local updates and extra mixing;

Future work:
• convergence analysis in the nonconvex case.

Thank you!

Communication-Efficient Distributed Optimization in Networks with Gradient Tracking
B. Li, S. Cen, Y. Chen, and Y. Chi, JMLR 2020.